UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/599,593	10/02/2006	Nikil Dutt	703538.4054	1321

34313 7590 10/18/2011

ORRICK, HERRINGTON & SUTCLIFFE, LLP
IP PROSECUTION DEPARTMENT
2050 Main Street, Suite 1100
IRVINE, CA 92614

EXAMINER	
WANG, JUE S	

ART UNIT	PAPER NUMBER
2193	

NOTIFICATION DATE	DELIVERY MODE
10/18/2011	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

IPPROSECUTION@ORRICK.COM
vsantos@orrick.com

Office Action Summary	Application No.	Applicant(s)	
	10/599,593	DUTT ET AL.	
	Examiner	Art Unit	
	JUE WANG	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 July 2011*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) *1-27* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1-27* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

12)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *7/20/2011*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-27 have been examined.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 2, 5, 9, 10, 24-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Leupers et al. "Generation of Interpretive and Compiled Instruction Set Simulators" (hereinafter Leupers), in view of Morley (US 5,781,758), further in view of Zemach et al. (US 7,107,580, hereinafter Zemach).

4. As per claim 1, Leupers teaches the invention as claimed, including a method of simulating an instruction set architecture (ISA) with a instruction set simulator (ISS), comprising:

fetching a first decoded instruction during a run time, wherein the decoded instruction is decoded from an original instruction in a target application program during a compile time preceding the run time (i.e., each instruction of a machine program is decoded at compile time, see page 340, Fig 1, page 342, section 4.3), the decoded instruction pointing to a template

configured to implement the functionality of the instruction (i.e., generating C macro calls to generic simulation functions, see page 341, section 4.1, 4.2), ; and

executing the designated template (see page 342, section 4.5).

Leupers does not explicitly teach wherein the template is associated with an instruction class that describes a set of instructions of the instruction set architecture having a common behavior, and wherein the original instruction is contained in the instruction class.

Morley teaches templates configured to implement the functionality of decoded instructions, wherein the template is associated with an instruction class that describes a set of instructions of the instruction set architecture having a common behavior, and wherein the original instruction is contained in the instruction class (i.e., associating one copy of a semantic routine with all different variations of the basic instruction, each having a unique encoding in the instruction set of the emulated code, see at least column 3, line 43 - column 5, line 19).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Leupers such that the template is associated with an instruction class that describes a set of instructions of the instruction set architecture having a common behavior, and wherein the original instruction is contained in the instruction class as taught by Morley to reduce the amount of storage space needed (see column 5, lines 50-68 of Morley).

Leupers and Morley do not teach determining whether the fetched instruction is modified from the original instruction and executing the designated template if the instruction was not modified.

Zemach teaches a method of binary translation that determines whether fetched instruction is modified from an original instruction and executes the translation of the fetched instruction if the instruction as not modified (see Fig 3, column 2, lines 51-64).

It would have been obvious to one of ordinary skill in the art at the time of the to have modified Leupers and Morley to determine whether the fetched instruction is modified from the original instruction and executing the designated template if the instruction was not modified as similarly taught by Zemach to detect self-modifying code which renders a previous translation of an instruction obsolete after the instruction is modified (see column 1, lines 64 - column 2, line 5, column 2, line 65 – column 3, line 10 of Zemach).

5. As per claim 2, Leupers teaches decoding the original instruction by selecting a template corresponding to the original instruction and customizing the template prior to fetching the instruction (see page 341, sections 4.1, 4.2).

Leupers does not explicitly teach that the customization is based on the data in original instruction.

Morley teaches customization of templates based on the data in original instruction (see at least Fig 3, column 4, line 49 – column 5, line 19).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Leupers such that the customization is based on data in the original instruction as taught by Morley to modify one copy of the template to accommodate different variations of the instruction to reduce the amount of storage space needed (see column 5, lines 50-68 of Morley).

6. As per claim 5, Morley teaches wherein the customizing the template comprises determining a value of a parameter in the template based on the data in the original instruction (see at least Fig 3, column 4, line 49 – column 5, line 19).

7. As per claim 9, Leupers does not teaches re-decoding the fetched instruction during the run time if the fetched instruction was modified, wherein the re-decoded instruction designates a function configured to implement the functionality of the instruction; and executing the designated function if the instruction was modified.

Zemach teaches re-decoding the fetched instruction during the run time if the fetched instruction was modified, wherein the re-decoded instruction designates a function configured to implement the functionality of the instruction; and executing the designated function if the instruction was modified (see Fig 3, Fig 4, column 1, lines 56-63, column 2, line 51-column 3, line 10).

It would have been obvious to one of ordinary skill in the art at the time of the to have modified Leupers to re-decode the fetched instruction during the run time if the fetched instruction was modified, wherein the re-decoded instruction designates a function configured to implement the functionality of the instruction; and execute the designated function if the instruction was modified as similarly taught by Zemach because self-modifying code must be re-translated since the modification renders a previous translation of an instruction obsolete (see column 1, lines 64 - column 2, line 5, column 2, line 65 – column 3, line 10 of Zemach).

8. As per claim 10, Zemach teaches executing the modified instructions using an interpretive process (see column 2, line 65 – column 3, line 10).

9. As per claims 24- 27, these are the computer readable medium claims of claims 1, 3, 9, and 10. Therefore, they are rejected using the same reasons as claims 1, 3, 9, and 10.

10. Claims 3, 4, 6-8, and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Leupers et al. "Generation of Interpretive and Compiled Instruction Set Simulators" (hereinafter Leupers), in view of Morley (US 5,781,758), further in view of Zemach et al. (US 7,107,580, hereinafter Zemach), further in view of DeWitt, Jr. et al. (US 2005/0102493 A1, hereinafter DeWitt).

11. As per claim 3, Leupers teaches wherein the template has a corresponding value usable to identify instructions belonging to an instruction class (i.e., case labels, see page 342, section 4.5).

Leupers does not teach the use of a corresponding mask usable to identify instructions belonging to an instruction class.

DeWitt is cited to teach the use of a mask usable to identify instructions belonging to an operation class (see at least [0189]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Leupers to use an operation mask usable to identify instructions belonging to an operation class as taught by DeWitt because the method chose to identify a type of instruction

is a design choice and the use of an operation mask is a well known way of identifying a type of instructions (see [0189] of DeWitt).

12. As per claim 4, Leupers teaches comparing the original instruction to the value corresponding to the template; and selecting the template if the value matches the original instruction (see page 342, section 4.5).

Leupers does not teach wherein the use of a mask.

DeWitt is cited to teach the use of a mask usable to identify instructions belonging to an operation class (see at least [0189]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Leupers to use a mask usable to identify instructions belonging to an operation class as taught by DeWitt because the method chose to identify a type of instruction is a design choice and the use of an operation mask is a well known way of identifying a type of instructions (see [0189] of DeWitt).

13. As per claim 6, Morley teaches wherein the customizing the template comprises determining a value of a parameter in the template based on the data in the original instruction (see at least Fig 3, column 4, line 49 – column 5, line 19).

14. As per claim 7, Leupers as modified teaches compiling a first program comprising the customized template in the compile time (see page 340, Figure 1, page 341, section 4.3).

15. As per claim 8, Leupers teaches optimizing the template during compile time (i.e., the simulation functions can be replaced by faster macros, see page 341, section 4.1).

16. As per claim 11, Leupers teaches compiling the target application program to generate the original instructions (see page 341, section 4.3).

17. Claims 12-16, 18, 19, 22, and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Killian et al. (US 6,477,683 B1, hereinafter Killian), in view of DeWitt, Jr. et al. (US 2005/0102493 A1, hereinafter DeWitt), further in view of Leupers et al. "Generation of Interpretive and Compiled Instruction Set Simulators" (hereinafter Leupers), further in view of Morley (US 5,781,758), further in view of Zemach et al. (US 7,107,580, hereinafter Zemach).

18. A system, comprising:

a processor (see column 9, line 56 - column 10, line 2); and

a computer readable medium having stored thereon a generic instruction model, the generic instruction model executable by the processor in an instruction set simulator (i.e., configuring a simulator for a specific instruction set architecture using a ISA definition, see Fig 6, column 6, line 65 – column 7, line 20, column 17, line 48 – column 18, line 1, column 31, lines 50-61), the generic instruction model comprising an instruction specification usable to interpret each instruction in an ISA, the instruction specification comprising one or more operation classes (i.e., the instruction class statement iclass, see column 14, line 10 - column 16, line 45);

wherein each operation class defines a set of one or more instructions (i.e., instruction semantic statement describes the behavior of one or more instructions, see column 16, lines 14-45), the operation class having a list usable to identify instructions belonging to the class (see column 16, lines 45-50); and

wherein the operation class comprises one or more symbols and an expression describing the class in terms of the one or more symbols (i.e., instruction semantic statement defines the opcode and operands of the instructions, the see column 16, lines 14-67), each symbol having a corresponding set of one or more symbol types, each symbol type in the set comprising information usable to determine the symbol when compared to an instruction (i.e., instruction operand statements identify registers and immediate constants, see at least column 14, lines 30-62, column 15, lines 14-36, column 16, lines 53-67); and

wherein the instruction set simulator is configured to perform

fetching a decoded instruction during a run time, and executing the decoded instruction (see at least column 31, lines 50-61).

Killian does not explicitly teach using an operation mask usable to identify instructions belonging to an operation class.

DeWitt is cited to teach the use of an operation mask usable to identify instructions belonging to an operation class (see at least [0189]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Killian to use an operation mask to identify instructions belonging to an operation class because it is well known in the art that the use of an operation mask is another

alternative to using a list to identify a type of instructions (see [0189] of DeWitt) and the use of a mask is well known to have the advantage of having a smaller storage requirement than a list.

Killian and DeWitt do not teach wherein the decoded instruction is decoded from an original instruction in a target application program during a compile time preceding the run time, the decoded instruction pointing to a template configured to implement the functionality of the instruction.

Leupers teaches a method of including a method of simulating an instruction set architecture (ISA) with a instruction set simulator (ISS), comprising: fetching a first decoded instruction during a run time, wherein the decoded instruction is decoded from an original instruction in a target application program during a compile time preceding the run time (i.e., each instruction of a machine program is decoded at compile time, see page 340, Fig 1, page 342, section 4.3), the decoded instruction pointing to a template configured to implement the functionality of the instruction (i.e., generating C macro calls to generic simulation functions, see page 341, section 4.1, 4.2), ; and executing the designated template (see page 342, section 4.5).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Killian and DeWitt such the decoded instruction is decoded from an original instruction in a target application program during a compile time preceding the run time, the decoded instruction pointing to a template configured to implement the functionality of the instruction as taught by Leupers to achieve higher execution speed by moving decoding overhead to simulator generation time (see page 339, right column, paragraph 3 of Leupers).

Killian, DeWitt, and Leupers do not explicitly teach wherein the template is associated with an instruction class that describes a set of instructions of the instruction set architecture

having a common behavior, and wherein the original instruction is contained in the instruction class.

Morley teaches templates configured to implement the functionality of decoded instructions, wherein the template is associated with an instruction class that describes a set of instructions of the instruction set architecture having a common behavior, and wherein the original instruction is contained in the instruction class (i.e., associating one copy of a semantic routine with all different variations of the basic instruction, each having a unique encoding in the instruction set of the emulated code, see at least column 3, line 43 - column 5, line 19).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Killian, DeWitt, and Leupers such that the template is associated with an instruction class that describes a set of instructions of the instruction set architecture having a common behavior, and wherein the original instruction is contained in the instruction class as taught by Morley to reduce the amount of storage space needed (see column 5, lines 50-68 of Morley).

Killian, DeWitt, Leupers and Morley do not teach determining whether the fetched instruction is modified from the original instruction and executing the designated template if the instruction was not modified.

Zemach teaches a method of binary translation that determines whether fetched instruction is modified from an original instruction and executes the translation of the fetched instruction if the instruction as not modified (see Fig 3, column 2, lines 51-64).

It would have been obvious to one of ordinary skill in the art at the time of the to have modified Killian, DeWitt, Leupers and Morley to determine whether the fetched instruction is

modified from the original instruction and executing the designated template if the instruction was not modified as similarly taught by Zemach to detect self-modifying code which renders a previous translation of an instruction obsolete after the instruction is modified (see column 1, lines 64 - column 2, line 5, column 2, line 65 – column 3, line 10 of Zemach).

19. As per claim 13, Killian teaches wherein the set of instructions has a common behavior and the expression defines the behavior of the class in terms of the one or more symbols (i.e., instruction semantic statement describes the behavior of one or more instructions, see column 16, lines 14-45).

20. As per claim 14, Killian teaches wherein one symbol type in the type set is a constant type (i.e., instruction operand statements operand identify registers and immediate constants, see at least column 15, lines 14-36).

21. As per claim 15, Killian teaches wherein the type set comprises a plurality of constant types, each constant type having a corresponding type mask usable to determine the constant when compared to an instruction (see column 14, lines 30-62, column 15, line 14 - column 16, line 13).

22. As per claim 16, Killian teaches wherein one symbol type in the type set is a register type (i.e., instruction operand statements operand identify registers and immediate constants, see at least column 15, lines 14-36).

23. As per claim 18, Killian teaches wherein one symbol type in the type set is an operation type (see column 14, lines 30-62, column 16, lines 53-67).

24. As per claim 19, Killian teaches wherein the type set comprises a plurality of operation types, each operation type having a corresponding type mask usable to determine the operation when compared to an instruction (i.e., the bits 20-23 in an instruction define the opcode, see column 14, lines 30-62, column 15, lines 1-5, column 16, lines 60-53-67).

25. As per claim 22, Killian teaches wherein each instruction comprise a series of binary data values (see at least column 14, lines 30-67).

DeWitt teaches the operation mask comprises a series of mask positions wherein each mask position corresponds to one instance of a binary data value (see at least [0189]).

26. As per claim 23, DeWitt teaches wherein each mask position has a value selected from a group comprising: a binary one value, a binary zero value and a do not care value (see [0189]).

27. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over Killian et al. (US 6,477,683 B1, hereinafter Killian), in view of DeWitt, Jr. et al. (US 2005/0102493 A1, hereinafter DeWitt), further in view of Leupers et al. "Generation of Interpretive and Compiled Instruction Set Simulators" (hereinafter Leupers), further in view of Morley (US 5,781,758), further in view of Zemach et al. (US 7,107,580, hereinafter Zemach), as applied to claim 16 above, further in view of Nohl et al. (US 2003/0217248 A1, hereinafter Nohl).

28. As per claim 17, Killian, DeWitt, Leupers, Morley, and Zemach do not explicitly teach wherein the register type comprises a register index and a register class.

Nohl teaches a processor description for an instruction that includes a register type comprising a register index and a register class (see Fig 5, [0050], [0051]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Killian, DeWitt, Leupers, Morley, and Zemach to include a register type comprising a register index and a register as taught by Nohl because the specification description for an instruction is a design choice.

29. Claims 20 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Killian et al. (US 6,477,683 B1, hereinafter Killian), in view of DeWitt, Jr. et al. (US 2005/0102493 A1, hereinafter DeWitt), further in view of Leupers et al. "Generation of Interpretive and Compiled Instruction Set Simulators" (hereinafter Leupers), further in view of Morley (US 5,781,758), further in view of Zemach et al. (US 7,107,580, hereinafter Zemach), as applied to claim 12 above, further in view of Wang et al. (US 2005/0160402 A1, hereinafter Wang).

30. As per claim 20, Killian teaches wherein the at least one operation class comprises a plurality of expressions (see column 16, line 40 - column 17, line 26).

Killian, DeWitt, Leupers, Morley, and Zemach do not explicitly teach wherein each expression being conditional on data within an instruction.

Wang teaches an operation class comprises a plurality of expressions where each expression conditional on data within an instruction (see [0110]-[0112]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Killian, DeWitt, Leupers, Morley, and Zemach to provide an operation class where each expression being conditional on data within an instruction as taught by Wang because the invention of Killian is usable to extend and customize the processor instruction set (see column 6, lines 43-64 of Killian) and it is advantageous to extend a processor instruction set with the conditional instruction as taught by Wang to reduce the number of operations that are performed (see [0110]-[0112] of Wang).

31. As per claim 21, Killian, DeWitt, Leupers, Morley, and Zemach do not explicitly teach wherein each instruction comprises a series of slots, each slot comprising data translatable into an operation.

Wang teaches an instruction specification wherein each instruction comprises a series of slots, each slot comprising data translatable into an operation (see [0035], [0036], [0055]-[0080]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Killian, DeWitt, Leupers, Morley, and Zemach to provide an instruction comprises a series of slots, each slot comprising data translatable into an operation as taught by Wang because the invention of Killian is usable to extend and customize the processor instruction set (see column 6, lines 43-64 of Killian) and it is advantageous to extend a processor instruction set to include instruction with a series of slots to allow parallel execution of operations (see [0007], [0035] of Wang).

Response to Arguments

32. Rejection of claims under §103(a):

33. As per claims 1-27, Applicants arguments have been fully considered but are moot in light of the new grounds of rejection.

Conclusion

34. Applicant's amendment necessitated the new ground(s) of rejection presented in this office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP §706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

35. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jue S. Wang whose telephone number is (571) 270-1655. The examiner can normally be reached on M, W-F 9:30 am - 5:00pm (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193

Jue Wang
Examiner
Art Unit 2193